

EKHARA Monte Carlo generator for $e^+e^- \rightarrow e^+e^-\pi^0$ and $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$ processes

Henryk Czyż^a, Sergiy Ivashyn^{a,b,*}

^a*Institute of Physics, University of Silesia, Uniwersytecka 4, Katowice PL-40007, Poland*

^b*NSC “KIPT”, Akademicheskaya 1, Kharkov UA-61108, Ukraine*

Abstract

We present EKHARA Monte Carlo event generator of reactions $e^+e^- \rightarrow e^+e^-\pi^0$ and $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$. The newly added channel ($e^+e^- \rightarrow e^+e^-\pi^0$) is important for $\gamma^*\gamma^*$ physics and can be used for the pion transition form factor studies at meson factories.

Keywords: Monte Carlo generator, Pion transition form factor, Pion pair production, Two-photon processes, e^+e^- annihilation

PROGRAM SUMMARY

Manuscript Title: EKHARA Monte Carlo generator for $e^+e^- \rightarrow e^+e^-\pi^0$ and $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$ processes

Authors: H. Czyż, S. Ivashyn

Program Title: EKHARA

Journal Reference:

Catalogue identifier:

Licensing provisions: none

Programming language: FORTRAN 77 with quadruple precision

Computer: PC, main frame

Operating system: Linux, Unix, MS Windows

RAM: up to 10 Megabytes for operation of the compiled program

Number of processors used: one

Keywords: Monte Carlo generator, Event simulation, Pion transition form factor, Pair production, Two-photon process, e^+e^- collision

Classification: 11.2 Phase Space and Event Simulation, 11.6 Phenomenological

*Corresponding author

Email address: ivashyn@kipt.kharkov.ua (Sergiy Ivashyn)

and Empirical Models and Theories

Nature of problem:

The first version of EKHARA [1,2] was developed to simulate background for the pion form factor measurement at meson factories coming from the process $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$. The newly added channel $e^+e^- \rightarrow e^+e^-\pi^0$ will help in the pion transition form factor studies at meson factories [3].

Solution method:

Events consisting of the momenta of the outgoing particles are generated by Monte Carlo methods. The generated events are distributed accordingly to the theoretical cross section. For the $e^+e^- \rightarrow e^+e^-\pi^0$ mode the Monte Carlo sampling developed in [4] was adopted.

Restrictions:

In order to compile the code, the FORTRAN 77 compiler should support quadruple precision numbers.

Unusual features:

Calculations are carried in quadruple precision, in order to avoid numerical cancellations in $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$ mode.

Running time:

Depends on the requested mode and applied kinematic cuts. Example: on Intel Core2 Quad CPU Q6600 @ 2.40GHz, using only one thread,

- 10^5 unweighted $e^+e^-\pi^0$ events are generated in 78 seconds (no cuts),
- 10^5 weighted $e^+e^-\pi^+\pi^-$ events are generated in 38 seconds (with cuts [2]).

References:

- [1] H. Czyz, E. Nowak-Kubat, Radiative return via electron pair production: Monte Carlo simulation of the process $e^+e^- \rightarrow \pi^+\pi^-e^+e^-$, Acta Phys. Polon., 2005, B36, 3425–3434
- [2] H. Czyz, E. Nowak-Kubat, The reaction $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$ and the pion form factor measurements via the radiative return method, Phys. Lett., 2006, B634, 493–497
- [3] G. Amelino-Camelia and others, Physics with the KLOE-2 experiment at the upgraded DAΦNE, Eur.Phys. J., 2010, C68, 619–681
- [4] G.A. Schuler, Two-photon physics with GALUGA 2.0, Comput. Phys. Commun., 1998, 108, 279–303

LONG WRITE-UP

1. Introduction

1.1. The physics case

The first version of EKHARA MC generator [1, 2] (EKHARA ver. 1) was developed to simulate the reaction $e^+e^- \rightarrow \pi^+\pi^-e^+e^-$. This process is a background [3] to the $e^+e^- \rightarrow \pi^+\pi^-\gamma(\gamma)$ cross section measurement at meson factories when only charged pions are observed [4, 5]. Its proper simulation is relevant for the precise extraction of the cross section $\sigma(e^+e^- \rightarrow \pi^+\pi^-)$ and the charged pion form factor using the *radiative return method*. The description of a physics content (matrix elements, modelling of the pion-photon interactions, etc.) in EKHARA ver. 1 was given in [2], however, the computational issues remained unpublished. In view of the accepted prolongation of the experiment, namely, the KLOE-2 project [6], and the planned radiative return program of the BES-III experiment [7, 8], this gap has to be filled and the description of the $e^+e^- \rightarrow \pi^+\pi^-e^+e^-$ mode of EKHARA generator is one of the aims of this paper. EKHARA ver. 1 is also a convenient base for an inclusion of other channels of the inelastic e^+e^- scattering.

The KLOE-2 project assumes an installation of special tagging devices [9], which will allow [6] to study, among other, the $e^+e^- \rightarrow \pi^0e^+e^-$ process, aiming at the following measurements:

- the two-photon decay width of π^0 ,
- the transition form factor $F_{\pi^0\gamma^*\gamma^*}(m_{\pi^0}^2, q_1^2, q_2^2)$ at space-like photon momentum transfers q_1^2, q_2^2 .

These measurements are of a significant importance [6] and the knowledge of $F_{\pi^0\gamma^*\gamma^*}$ for high photon virtualities is supposed to help in the reduction of an error in the calculation of the light-by-light contributions to the muon anomalous magnetic moment. A reliable Monte Carlo generator for this channel, based on the modern knowledge of the $\pi^0 - \gamma^* - \gamma^*$ transition form factor is an indispensable tool for such studies. A description of this new mode, which is distributed within EKHARA ver. 2, is the second aim of this paper.

1.2. Basic functionalities of EKHARA ver. 2

- $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$
 - generates weighted events;
 - fills the histograms;
 - gives the integrated cross section within cuts.
- $e^+e^- \rightarrow e^+e^-\pi^0$
 - generates and stores unweighted events;
 - fills the histograms;
 - gives the integrated cross section within cuts.

The program is supplemented with Gnuplot scripts for visualization of the histograms, produced by EKHARA.

1.3. Related Monte Carlo programs

There exist several MC generators for $\gamma\gamma$ physics:

1. the code written by A. Courau [10]; used in [11, 12] for the $e^+e^- \rightarrow e^+e^-\pi^0\pi^0$, and in [13] for the $e^+e^- \rightarrow e^+e^-\pi^0$ process;
2. the code by F. Nguyen *et al.* [14] for $e^+e^- \rightarrow e^+e^-\pi^0\pi^0$;
3. TREPS program written by S. Uehara [15] and used by Belle collaboration [16, 17];
4. TWO GAM developed by D. M. Coffman, used by CLEO [18] for the $e^+e^- \rightarrow e^+e^-P$, with $P = \pi^0, \eta, \eta'$;
5. GGRS RC used by the BaBar collaboration [19];
6. GALUGA by G. A. Schuler [20] for LEP2 physics;
7. GaGaRes written by F. A. Berends and R. van Gulik [21], for the study of resonance production in $\gamma\gamma$ interaction at LEP2 energies.

GALUGA and GaGaRes generators describe high-energy physics and adapting them to much lower energies is not straightforward. Other programs are not public. Moreover, the Equivalent Photon Approximation (EPA) is employed to a large extent in the majority of the generators. EPA is a useful simplification in the description of the two photon processes, when the accuracy requirements are not high. It leads however to some discrepancies with respect to the exact formulation, as shown already in [22], especially for single-pseudoscalar final states.

Summarizing, the basic requirements for a MC generator for $e^+e^- \rightarrow e^+e^-P$ studies not restricted to the region where the photons are quasi-real, are:

1. use exact formulae and exact kinematics (do not use EPA),
2. include both s - and t -channel amplitudes and their interference,
3. allow user-defined form factors,
4. implement specific kinematic cuts,
5. account for the peaking behaviour of the cross section, in order to have a good Monte Carlo efficiency.

The EKHARA ver. 2, presented in this paper, fulfils all the listed above criteria.

2. The generation of four-momenta for the one-pion mode

The differential cross-section for the reaction $e^+e^- \rightarrow e^+e^-\pi^0$, averaged over helicities of the initial e^+e^- states, is given by

$$d\sigma_{avg}(e^+e^- \rightarrow e^+e^-\pi^0) = \frac{1}{4} \frac{1}{2s} d\text{Lips}_3 \sum |\mathcal{M}_{\pi^0}|^2. \quad (1)$$

Here s is the initial electron-positron invariant mass squared, $1/4$ is the averaging factor, $2s$ is the flux factor. By $d\text{Lips}_n$ we denote a differential n -body Lorentz-invariant phase space and \mathcal{M}_{π^0} stands for the matrix element describing the reaction $e^+e^- \rightarrow e^+e^-\pi^0$. We generate the kinematics in the center-of-mass frame of the initial e^+e^- , with the z -axis along the initial positron momentum. The event is determined by five kinematic invariants and one azimuthal angle ϕ :

$$\begin{aligned} d\sigma(e^+(p_1) e^-(p_2) \rightarrow e^+(q_1) e^-(q_2) \pi^0(Q)) &= d\sigma(s; t_1, t_2, s_1, s_2, \phi), \\ s &= (p_1 + p_2)^2, \\ t_1 &= (p_1 - q_1)^2, \\ t_2 &= (p_2 - q_2)^2, \\ s_1 &= (p_1 + q_2)^2, \\ s_2 &= (p_2 + q_1)^2. \end{aligned} \quad (2)$$

2.1. The matrix element

The matrix element for the reaction $e^+e^- \rightarrow e^+e^-P$ at the tree level contains the t -channel and the s -channel parts depicted in Fig. 1 ($\mathcal{M}_{\pi^0} = \mathcal{M}_t + \mathcal{M}_s$).

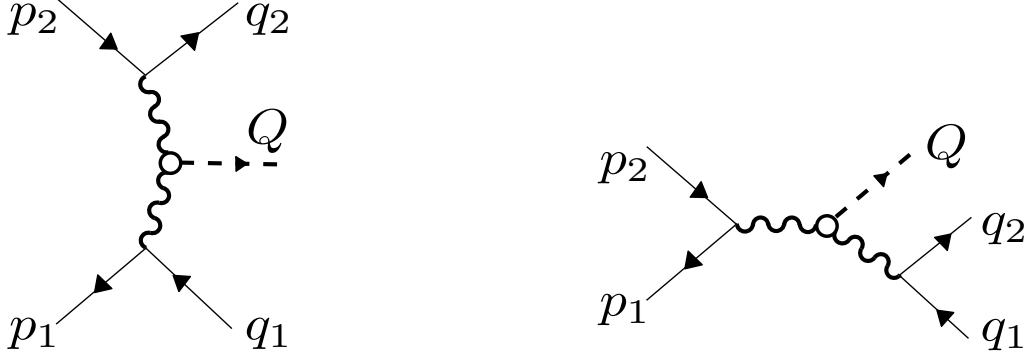


Figure 1: The t -channel (*left*) and the s -channel (*right*) diagrams for $e^+e^- \rightarrow e^+e^- P$

The t -channel matrix element has the following form

$$\begin{aligned} \mathcal{M}_t = & -\frac{4i\alpha^2}{f_\pi} F(t_1, t_2) \epsilon_{\mu\nu\alpha\beta} \frac{1}{t_1 t_2} (q_1 - p_1)^\alpha (q_2 - p_2)^\beta \\ & \times (\bar{v}(p_1) \gamma^\mu v(q_1)) (\bar{u}(q_2) \gamma^\nu u(p_2)). \end{aligned} \quad (3)$$

The completely antisymmetric tensor $\epsilon^{\mu\nu\alpha\beta}$ is defined by $\epsilon_{0123} = -\epsilon^{0123} = 1$. The t -channel contribution to the cross section, is highly peaked at small t_1 and/or t_2 values. The pion two-photon transition form factor $F(t_1, t_2)$ is an important ingredient, see, e.g., [23] and it provides an additional dumping of the amplitude at large values of t_1, t_2 . The normalization is $F(0, 0) = 1$, $\alpha \approx 1/137$ is the fine structure constant and $f_\pi \approx 92.4$ MeV is the pion decay constant.

In Fig. 2 we demonstrate an agreement of the Monte Carlo simulation with the “single-tag” experimental data from CLEO [18] and BaBar [19]. For simulation we use the matrix element (3) with the form factor of the lowest meson dominance model with two vector resonances (LMD+V), fitted [24] to the BABAR data [19]. A number of other expressions for the form factor are also available for the user of EKHARA. For example, one can use the formulae given by the simple vector meson dominance ansatz (rho meson pole), or the lowest meson dominance approach results with one vector resonance multiplet (LMD) or two multiplets (LMD+V) with the generic form obtained from the operator product expansion considerations [23]. One may also use the form factor derived in a quark model, which resembles a reasonable agreement with the BaBar [19] data at high momentum transfer [25].

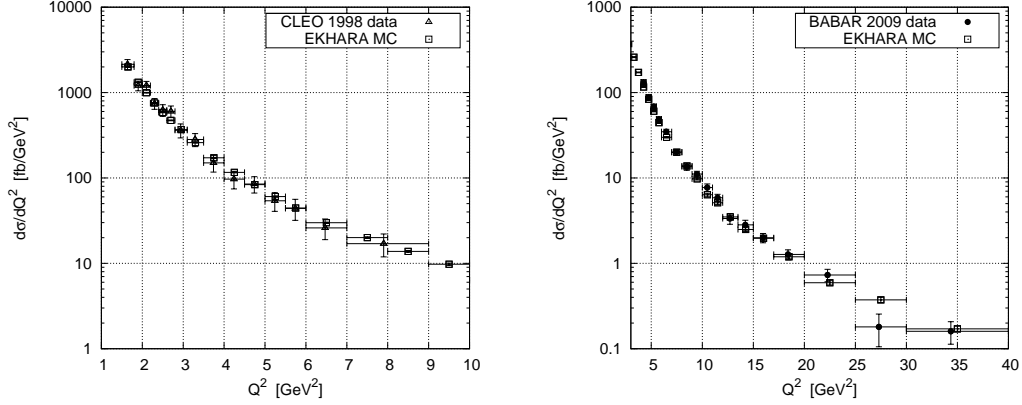


Figure 2: Comparison of EKHARA Monte Carlo simulation with experimental data from CLEO [18] and BaBar [19]

The matrix element for the s -channel reads

$$\mathcal{M}_s = \frac{4i\alpha^2}{f_\pi} F(s, (q_1 + q_2)^2) \epsilon_{\mu\nu\alpha\beta} \frac{1}{s (q_1 + q_2)^2} (p_1 + p_2)^\alpha (q_1 + q_2)^\beta (\bar{v}(p_1) \gamma^\mu u(p_2)) (\bar{u}(q_2) \gamma^\nu v(q_1)). \quad (4)$$

The form factor $F(s, (q_1 + q_2)^2)$ in the s -channel has to be given by the same analytic function as in the t -channel case. The s -channel contribution, having a peak at small invariant masses of the final e^+e^- pair, is much smaller than the t -channel one when at least one of the t_1, t_2 invariants is small and it is often missing in the existing MC codes. In EKHARA all the amplitudes are included (s - and t -channel as well as their interference). This guaranties that no relevant piece is missing even for the kinematic configurations where both t_1 and t_2 are large.

The peaking behaviour of the s - and t -channel amplitudes is different. Therefore EKHARA uses, depending on the set of amplitudes included in the calculation ($\mathcal{M}_s, \mathcal{M}_t$ or both), one of the three generation procedures described below.

2.2. t -channel

To generate the particle four-momenta we have adopted a very efficient generating procedure, which was used in the MC generator GALUGA [20]. Here we only sketch its layout and for further details we refer the reader to

[20]. The Lorentz-invariant phase space is mapped to a unit hypercube in the space of the uniformly distributed random numbers $r_i \in [0, 1]$, $i = 1, \dots, 5$:

$$\mathrm{d} \text{Lips}_3 = V^t \mathrm{d} t_1 \mathrm{d} t_2 \mathrm{d} s_1 \mathrm{d} s_2 \mathrm{d} \phi = V^t J_t \prod_{i=1}^5 \mathrm{d} r_i , \quad (5)$$

where J_t is the mapping Jacobian and the volume factor reads

$$V^t = \frac{4\pi^2}{\pi^5} \frac{1}{2^9 s \beta} , \quad (6)$$

with

$$\beta = 1 - 4m_e^2/s . \quad (7)$$

The electron mass is denoted by m_e .

The user is supposed to provide the following cuts (minima and maxima):

- for t_2 : CUT_t2min and CUT_t2max,
- for t_1 : CUT_t1min and CUT_t1max,
- for positron polar angle: CUT_th1min and CUT_th1max,
- for electron polar angle: CUT_th2min and CUT_th2max,
- for positron energy: E1_min and E1_max,
- for electron energy: E2_min and E2_max.

The kinematic invariants are generated in the following sequence:

1. calculate cuts on t_2
 - t2max according to CUT_th2min and E2_min, E2_max
 - t2min according to CUT_th2max and E2_min, E2_max
 - correct t2max and t2min according to CUT_t2min and CUT_t2max
2. generate t_2 ; t2min $\leq t_2 \leq$ t2max
3. calculate cuts on t_1
 - t1max according to CUT_th1min and E1_min, E1_max
and according to generated t_2 (the lowest is taken)

- **t1min** according to **CUT_th1max** and **E1_min**, **E1_max**
and according to generated t_2 (the highest is taken)
 - correct **t1max** and **t1min** according to **CUT_t1min** and **CUT_t1max**
4. generate t_1 ; **t1min** $\leq t_1 \leq$ **t1max**
 5. generate s_1
 6. generate s_2

The changes of variables are the following:

$$t_2 = \text{t2min} \exp \left(r_1 \ln \frac{\text{t2max}}{\text{t2min}} \right), \quad (8)$$

$$t_1 = \text{t1min} \exp \left(r_2 \ln \frac{\text{t1max}}{\text{t1min}} \right), \quad (9)$$

$$s_1 = \frac{X_1}{2} + m_e^2 + t_2 + 2m_e^2 \frac{t_2}{X_1}, \quad (10)$$

$$s_2 = \frac{-b + \sqrt{\Delta} \sin(\pi(r_4 - 1/2))}{2a}, \quad (11)$$

where

$$X_1 = (\nu + W)(1 + y_1) \exp(r_3 \delta), \quad (12)$$

$$\delta = \ln \frac{s(1 + \beta)^2}{(\nu + W)(1 + y_1)(1 + y_2)}, \quad (13)$$

$$W = \sqrt{\nu^2 - t_1 t_2}, \quad (14)$$

$$\nu = (m_\pi^2 - t_1 - t_2)/2, \quad (15)$$

$$y_{1,2} = \sqrt{1 - 4m_e^2/t_{1,2}}, \quad (16)$$

where m_π stands for the π^0 mass. Let Δ_4 be the 4×4 symmetric Gram determinant of any four independent vectors formed out of p_1 , p_2 , q_1 , Q , q_2 , then its expansion in powers of s_2 determines the coefficients a , b and c : $16\Delta_4 \equiv as_2^2 + bs_2 + c$, we also use $\Delta = b^2 - 4ac$. For numerically stable forms of Δ , a , b and c we refer to [20]. From t_1 , t_2 , s_1 , s_2 one can calculate the final state particles four-momenta in a frame where the $x-z$ plane is given by the initial positron and final π^0 momenta. Subsequently, the ‘event’ is randomly rotated around the z -axis [20] using the r_5 random number.

For this mapping one has

$$J_t = \delta \ln(\text{t1max}/\text{t1min}) \ln(\text{t2max}/\text{t2min}) D_t \equiv \tilde{J}_t D_t, \quad (17)$$

$$D_t = t_1 t_2. \quad (18)$$

The D_t factor absorbs the peaking behaviour of the t -channel amplitude. The value of $\tilde{J}_t \equiv J_t/D_t$ for each given event is stored in the variable **JacobianFactor**. The differential cross section with the t -channel mapping reads

$$d\sigma_{avg} = \frac{1}{4} \frac{1}{2s} V^t \tilde{J}_t \prod_{i=1}^5 dr_i \left(D_t \sum |\mathcal{M}|^2 \right). \quad (19)$$

The above procedure is used when user requires to use the \mathcal{M}_t amplitude only.

2.3. s -channel

The Lorentz-invariant phase space $d\text{Lips}_3$ in eq. (1) can be factorized in the following way:

$$d\text{Lips}_3 = \frac{dk^2}{2\pi} d\text{Lips}_2^i d\text{Lips}_2^f, \quad (20)$$

where k^2 has a natural meaning of the invariant mass squared of the virtual photon, which couples to the final e^+e^- in the s -channel diagram.

Explicit expressions for these 2-body phase spaces are

$$d\text{Lips}_2^i = \frac{1}{(2\pi)^2} \frac{1}{4\sqrt{s}} \sqrt{\frac{(s+k^2-m_\pi^2)^2}{4s} - k^2} d\Omega_i, \quad (21)$$

$$d\text{Lips}_2^f = \frac{1}{(2\pi)^2} \frac{1}{4\sqrt{k^2}} \sqrt{\frac{k^2}{4} - m_e^2} d\Omega_f^*, \quad (22)$$

where $d\Omega_f^*$ is the differential solid angle in the center of mass frame of final e^+e^- , i.e. the self frame of the k^2 and $d\Omega_i$ denotes the pion solid angle in the center of mass frame of the initial e^+e^- pair. Thus, (20) takes the form

$$d\text{Lips}_3 = V^s dk^2 d\Omega_i d\Omega_f^*, \quad (23)$$

where the volume factor reads

$$V^s = \frac{1}{29\pi^5} \frac{1}{\sqrt{sk^2}} \sqrt{\frac{k^2}{4} - m_e^2} \sqrt{\frac{(s+k^2-m_\pi^2)^2}{4s} - k^2}. \quad (24)$$

The Lorentz-invariant phase space is mapped to a unit hypercube in the space of the uniformly distributed random numbers $r_i \in [0, 1]$, $i = 1, \dots, 5$:

$$\mathrm{d} \text{Lips}_3 = V^s J_s \prod_{i=1}^5 \mathrm{d} r_i , \quad (25)$$

where J_s is the Jacobian for a mapping from physical variables to r_i .

The following mapping is adopted:

$$\begin{aligned} \phi_i &= 2\pi r_1 , \\ \cos \theta_i &= -1 + 2r_2; \quad \sin \theta_i = 2\sqrt{r_2(1-r_2)} , \\ \phi_f^* &= 2\pi r_3 , \\ \cos \theta_f^* &= -1 + 2r_4; \quad \sin \theta_f^* = 2\sqrt{r_4(1-r_4)} , \end{aligned} \quad (26)$$

and the invariant mass squared (k^2) of the final e^+e^- pair is generated using logarithmic mapping

$$k^2 = 4m_e^2 \left(\frac{(\sqrt{s} - m_\pi)^2}{4m_e^2} \right)^{r_5} . \quad (27)$$

From (26) and (27) one can recover the momenta of all final state particles:

$$\begin{aligned} |\vec{Q}| &= \frac{1}{2\sqrt{s}} \sqrt{\lambda(s, k^2, m_\pi^2)} , \\ \vec{Q} &= |\vec{Q}| \begin{pmatrix} \sin \theta_i \sin \phi_i \\ \sin \theta_i \cos \phi_i \\ \cos \theta_i \end{pmatrix} , \\ k &\equiv (E_k, \vec{k}) = (\sqrt{k^2 + |\vec{Q}|^2}, -\vec{Q}) , \end{aligned} \quad (28)$$

and

$$\begin{aligned} |\vec{q}_2^*| &= \frac{1}{2\sqrt{k^2}} \sqrt{\lambda(k^2, m_e^2, m_e^2)} , \\ \vec{q}_2^* &= |\vec{q}_2^*| \begin{pmatrix} \sin \theta_f^* \sin \phi_f^* \\ \sin \theta_f^* \cos \phi_f^* \\ \cos \theta_f^* \end{pmatrix} , \\ q_2^* &= (\sqrt{m_e^2 + |\vec{q}_2^*|^2}, \vec{q}_2^*) , \\ q_1^* &= (\sqrt{m_e^2 + |\vec{q}_2^*|^2}, -\vec{q}_2^*) . \end{aligned} \quad (29)$$

The kinematic variables with asterisks are given in the center of mass frame of the final e^+e^- pair, i.e. in the self frame of the second virtual photon. In order to obtain the values of the final positron and electron momenta in the lab frame, the q_1 and q_2 , a Lorentz transformation of q_1^* and q_2^* is performed.

For this mapping, we have

$$J_s = 4(2\pi)^2 \ln \left(\frac{(\sqrt{s} - m_\pi)^2}{4m_e^2} \right) D_s \equiv \tilde{J}_s D_s, \quad (30)$$

$$D_s = k^2. \quad (31)$$

The D_s factor absorbs the peaking behaviour of the s -channel amplitude. The value of $\tilde{J}_s \equiv J_s/D_s$ for each given event is stored in the variable **JacobianFactor**. The differential cross section with the s -channel mapping reads

$$d\sigma_{avg} = \frac{1}{4} \frac{1}{2s} V^s \tilde{J}_s \prod_{i=1}^5 dr_i \left(D_s \sum |\mathcal{M}|^2 \right). \quad (32)$$

The above procedure is used when user requires to use the \mathcal{M}_s amplitude only.

2.4. Merging s - and t -channels

The differential cross section can be written as

$$d\sigma_{avg} = \frac{1}{4} \frac{1}{2s} \prod_{i=1}^5 dr_i V J \left(\frac{A/D_s + B/D_t}{A/D_s + B/D_t} \right) \sum |\mathcal{M}|^2 \quad (33)$$

$$= \frac{1}{4} \frac{1}{2s} \prod_{i=1}^5 dr_i \left(A V^s \tilde{J}_s + B V^t \tilde{J}_t \right) \frac{D_s D_t}{B D_s + A D_t} \sum |\mathcal{M}|^2 \quad (34)$$

$$= \frac{1}{4} \frac{1}{2s} \int_0^A dr_6 \prod_{i=1}^5 dr_i V^s \tilde{J}_s \frac{D_s D_t}{B D_s + A D_t} \sum |\mathcal{M}|^2 \\ + \frac{1}{4} \frac{1}{2s} \int_A^1 dr_6 \prod_{i=1}^5 dr_i V^t \tilde{J}_t \frac{D_s D_t}{B D_s + A D_t} \sum |\mathcal{M}|^2, \quad (35)$$

where $r_i \in [0, 1]$, $i = 1, \dots, 5$ are the uniform random numbers, and $A \equiv (1-r_m)$, $B \equiv r_m$, with an a priori weight $r_m \in (0, 1)$. We find empirically that

$r_m = 0.9$ gives an efficient merging in order to generate events distributed according to $|M_s + M_t|^2$. The s-channel (t-channel) is used with a probability $1 - r_m$ (r_m). In each channel procedures described in Sections 2.2 and 2.3) are used.

The above procedure is followed when user requires to use the full amplitude $\mathcal{M} = \mathcal{M}_s + \mathcal{M}_t$.

2.5. Generation of the unweighted events

Let the UB be a pre-evaluated upper bound for the Monte Carlo integrand *Contrib*. For generation of the unweighted events, we use the following accept–reject method:

1. calculate *Contrib*,
2. generate random number r_{accept} ,
3. accept event, if $Contrib \geq r_{accept} UB$.

The explicit expression for *Contrib* depends on the mapping, which is used by the generator, and reads

- s–channel

$$Contrib = \frac{1}{4} V^s J_s \sum |\mathcal{M}|^2,$$

- t–channel

$$Contrib = \frac{1}{4} V^t J_t \sum |\mathcal{M}|^2,$$

- Merged s– and t–channel

$$Contrib = \begin{cases} \frac{1}{4} V^s \tilde{J}_s \frac{D_s D_t}{BD_s + AD_t} \sum |\mathcal{M}|^2 & \text{with probability } (1 - r_m), \\ \frac{1}{4} V^t \tilde{J}_t \frac{D_s D_t}{BD_s + AD_t} \sum |\mathcal{M}|^2 & \text{with probability } (r_m). \end{cases}$$

The flowchart for Monte Carlo generator routine is given in Figure 3. It illustrates the order of phase space generation, application of cuts and the use of accept/reject method. In the block of the numerical stability control we make sure that the upper bound of the integrand is not overshooted, the energy and momentum conservation holds true, the final particles have on-shell momenta, the Gramm determinants are positively defined and that the generated sines and cosines of the particle angles are within $[-1, 1]$.

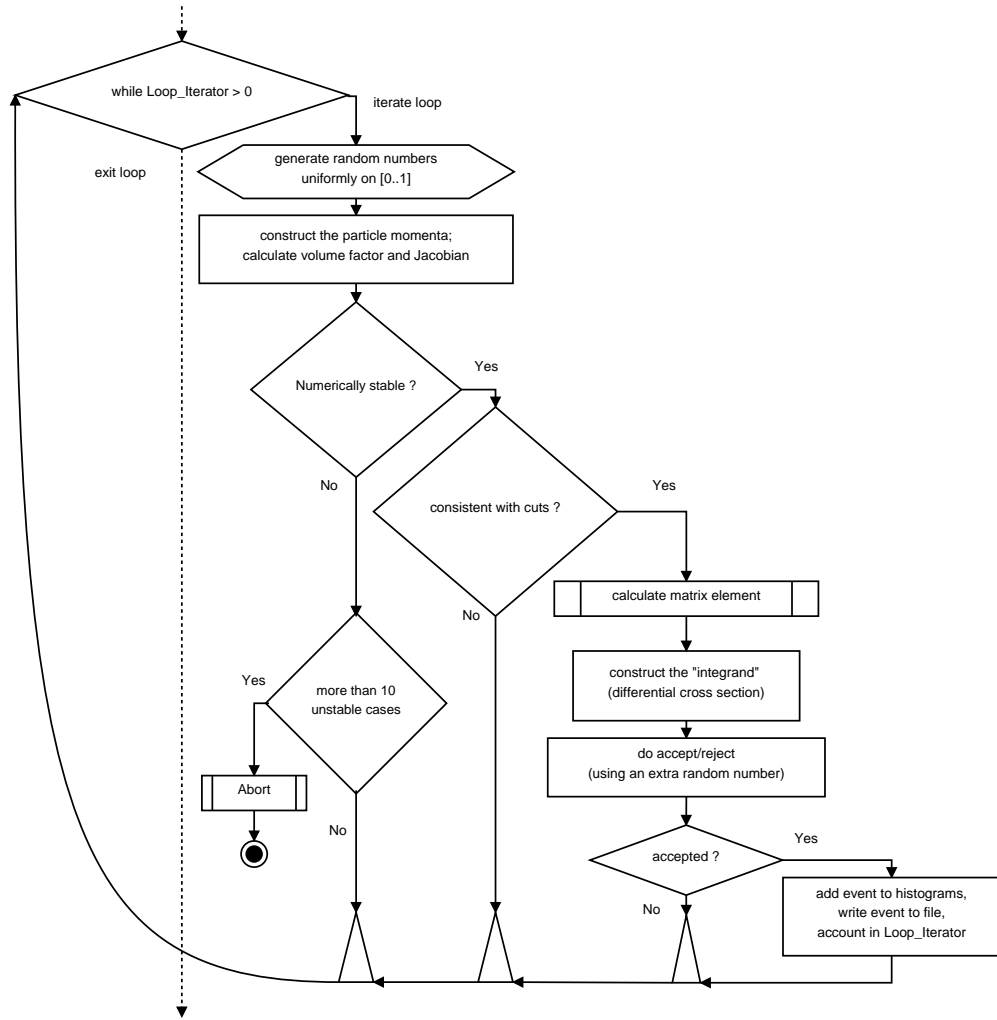


Figure 3: Flowchart for the $e^+e^- \rightarrow e^+e^-\pi^0$ event generation.

3. The generation of four-momenta in the two-pions mode

The differential cross-section for the reaction $e^+(p_1)e^-(p_2) \rightarrow \pi^+(\pi_1)\pi^-(\pi_2)e^+(q_1)e^-(q_2)$, averaged over helicity states of the initial e^+e^- , is given by

$$d\sigma_{avg}(e^+e^- \rightarrow \pi^+\pi^-e^+e^-) = \frac{1}{4} \frac{1}{2s} d\text{Lips}_4 \sum |\mathcal{M}_{\pi^+\pi^-}|^2. \quad (36)$$

The explicit expression for the matrix element $\mathcal{M}_{\pi^+\pi^-}$ can be found in [2] and its numerical evaluation is carried out in the helicity amplitudes framework.

The multi-channel variance reduction method is used to improve efficiency of the generator and the generation is split into four channels, where two of them absorb peaks present in t -channel diagrams and other two take care of the s -channel peaks. At difference to one-pion mode only one procedure for generation is used independently on the included contributions. This guaranties an efficient generation when s - and t -channel contributions are summed in the matrix element. Moreover, as this mode was constructed for generation of a background for the radiative return events, the generator is not optimised to run in the region, where $\gamma^*\gamma^*$ contributions dominate.

In this Section we use the following definitions:

$$k_1 = q_1 + q_2, \quad Q = \pi_1 + \pi_2, \quad (37)$$

$$s = (p_1 + p_2)^2, \quad t = (q_2 - p_2)^2, \quad t_1 = (p_1 - q_1)^2. \quad (38)$$

3.1. t -channel

For the t -channel peaks absorption, we use the following phase space representation

$$d\text{Lips}_4(p_1 + p_2; q_1, q_2, \pi_1, \pi_2) = d\text{Lips}_2(p_1 + p_2; Q', q_2) \frac{dQ'^2}{2\pi} d\text{Lips}_2(Q'; Q, q_1) \frac{dQ^2}{2\pi} d\text{Lips}_2(Q; \pi_1, \pi_2) \quad (39)$$

in one of the channels and an analogous one, with $q_1 \leftrightarrow q_2$, in the other channel. As both channels are completely symmetric under $q_1 \leftrightarrow q_2$, we will describe here only changes of variables, which smoothen the distribution, only in one of them. For the two invariant masses (Q^2 and Q'^2) the following change of variables was performed

$$Q^2 = (\sqrt{s} - 2m_e - \sqrt{-2z})^2, \quad z = -\frac{1}{2}(\sqrt{s} - 2m_e - 2m_\pi)^2(1 - r_{Q^2}), \quad (40)$$

$$Q'^2 = \frac{1}{3\sqrt{-3y}} + m_e^2, \quad (41)$$

$$y = -\frac{1}{3(Q^2 + 2\sqrt{Q^2}m_e)^3} + \left(\frac{1}{3(Q^2 + 2\sqrt{Q^2}m_e)^3} - \frac{1}{3(s - 2\sqrt{sm_e})^3} \right) r_{Q'^2}.$$

The angles of \vec{q}_2 vector are defined in the initial e^+e^- center of mass (cms) frame with z-axis along \vec{p}_1 and the polar angle is used to absorb the peak coming from the propagator of the photon exchanged in the t -channel

$$\begin{aligned} \cos \theta_{q_2} &= \frac{3m_e^2 - s + Q'^2 - 2t}{\sqrt{1 - \frac{4m_e^2}{s}\lambda^{1/2}(s, Q'^2, m_e^2)}}, \quad t = -\frac{1}{y}, \\ y &= -\frac{1}{t_-} + \frac{s\sqrt{1 - \frac{4m_e^2}{s}\lambda^{1/2}(s, Q'^2, m_e^2)}}{m_e^2(Q'^2 - m_e^2)^2} r_{\theta_{q_2}}, \quad \phi_{q_2} = 2\pi r_{\phi_{q_2}}, \end{aligned} \quad (42)$$

where

$$t_- = \frac{1}{2} \left(3m_e^2 - s + Q'^2 - \sqrt{1 - \frac{4m_e^2}{s}\lambda^{1/2}(s, Q'^2, m_e^2)} \right). \quad (43)$$

The angles of the \vec{Q} vector are defined in Q' rest frame and the appropriate change of variables reads

$$\begin{aligned} \cos \theta_Q &= \frac{2E'_1 Q_0 - Q^2 - \frac{1}{2|\vec{p}_1| |\vec{Q}|_x}}{2|\vec{p}_1| |\vec{Q}|}, \quad \phi_Q = 2\pi r_{\phi_Q} \\ x &= \frac{-1}{2|\vec{p}_1| |\vec{Q}| (Q^2 - 2E'_1 Q_0 - 2|\vec{p}_1| |\vec{Q}|)} + \frac{2}{(Q^2 - 2E'_1 Q_0)^2 - 4|\vec{p}_1|^2 |\vec{Q}|^2} r_{\theta_Q}, \end{aligned} \quad (44)$$

where $E'_1 = \frac{Q'^2 - t + m_e^2}{2\sqrt{Q'^2}}$, $Q_0 = \frac{Q'^2 + Q^2 - m_e^2}{2\sqrt{Q'^2}}$, $|\vec{p}_1| = \frac{\lambda^{1/2}(Q'^2, t, m_e^2)}{2\sqrt{Q'^2}}$, $|\vec{Q}| = \frac{\lambda^{1/2}(Q'^2, Q^2, m_e^2)}{2\sqrt{Q'^2}}$ and $\lambda(a, b, c) = a^2 + b^2 + c^2 - 2(ab + ac + bc)$. As we chose

here the z-axis along the \vec{p}_1 (the \vec{p}_1 in the Q' rest frame) the vectors are rotated after generation to restore the general choice of the z-axis along \vec{p}_1 in the e^+e^- cms frame.

Finally the angles of the positively charged pion are generated in the Q rest frame with flat distributions

$$\cos \theta_{\pi_1} = -1 + 2r_{\theta_{\pi_1}} , \quad \phi_{\pi_1} = 2\pi r_{\phi_{\pi_1}} . \quad (45)$$

The described change of variables transforms the phase space into a unit hypercube ($0 < r_i < 1$, $i = Q^2, \dots, \phi_{\pi_1}$) and collecting all the jacobians it reads

$$d \text{Lips}_4(p_1 + p_2; q_1, q_2, \pi_1, \pi_2) = P(q_1, q_2) dr_{Q^2} dr_{Q'^2} dr_{\theta_{q_2}} dr_{\phi_{q_2}} dr_{\theta_Q} dr_{\phi_Q} dr_{\theta_{\pi_1}} dr_{\phi_{\pi_1}} \quad (46)$$

with

$$\begin{aligned} P(q_1, q_2) = & \frac{1}{6(4\pi)^5 Q'^2 m_e^2} \lambda^{1/2}(Q'^2, Q^2, m_e^2) \lambda^{1/2}(s, Q'^2, m_e^2) \sqrt{1 - \frac{4m_\pi^2}{Q^2}} \\ & \frac{t^2 (Q'^2 - m_e^2)^2 (Q^2 - 2Q \cdot p_1)^2}{(Q^2 - 2E'_1 Q_0)^2 - 4|\vec{p}'_1|^2 |\vec{Q}|^2} \frac{\sqrt{Q^2}(\sqrt{s} - 2m_e - 2m_\pi)^2}{\sqrt{s} - 2m_e - \sqrt{Q^2}} \\ & \left(\frac{1}{(Q^2 + 2\sqrt{Q^2} m_e)^3} - \frac{1}{(s - 2\sqrt{s} m_e)^3} \right) . \end{aligned} \quad (47)$$

3.2. *s-channel*

For the *s*-channel generation it is convenient to write the phase space in the following form

$$\begin{aligned} d \text{Lips}_4(p_1 + p_2; q_1, q_2, \pi_1, \pi_2) = \\ d \text{Lips}_2(p_1 + p_2; Q, k_1) \frac{dk_1^2}{2\pi} d \text{Lips}_2(k_1; q_1, q_2) \frac{dQ^2}{2\pi} d \text{Lips}_2(Q; \pi_1, \pi_2). \end{aligned} \quad (48)$$

The two generation channels used here differ only in the generation of the electron-positron pair invariant mass k_1^2 and the change of variables will be described simultaneously. The invariant mass Q^2 is generated with a flat distribution

$$Q^2 = 4m_\pi^2 + ((\sqrt{s} - 2m_e)^2 - 4m_\pi^2) r_{Q^2} . \quad (49)$$

Reflecting two leading k_1^2 behaviors of the cross section, the two distinct changes of variables are done in the generation of k_1^2 :

$$\begin{aligned} k_1^2 &= s \exp(y_I^{1/3}) , \\ y_I &= \ln^3(4m_e^2/s) + \left[\ln^3 \left(\left(1 - \sqrt{Q^2/s} \right)^2 \right) - \ln^3(4m_e^2/s) \right] r_{k_1^2, I} \end{aligned} \quad (50)$$

$$\begin{aligned} k_1^2 &= s (1 - \exp(-y_{II})) , \\ y_{II} &= -\ln(1 - 4m_e^2/s) - \ln \left(\frac{\sqrt{Q^2} (2\sqrt{s} - \sqrt{Q^2})}{(s - 4m_e^2)} \right) r_{k_1^2, II} . \end{aligned} \quad (51)$$

The \vec{k}_1 polar angle is used to absorb peaks coming from the electron propagator, while its azimuthal angle is generated with a flat distribution:

$$\begin{aligned} \phi_{k_1} &= 2\pi r_{\phi_{k_1}} , \quad \cos \theta_{k_1} = \frac{-k_1^2 + 2k_{10}p_{10}}{2|\vec{k}_1||\vec{p}_1|} \tanh \left(\frac{y}{2} \right) \\ y &= \ln \left(\frac{k_1^2 - 2k_{10}p_{10} + 2|\vec{k}_1||\vec{p}_1|}{k_1^2 - 2k_{10}p_{10} - 2|\vec{k}_1||\vec{p}_1|} \right) + \ln \left(\frac{k_1^2 - 2k_{10}p_{10} - 2|\vec{k}_1||\vec{p}_1|}{k_1^2 - 2k_{10}p_{10} + 2|\vec{k}_1||\vec{p}_1|} \right)^2 r_{\theta_{k_1}} , \end{aligned} \quad (52)$$

where $k_{10} = \frac{s+k_1^2-Q^2}{2\sqrt{s}}$, $p_{10} = \frac{\sqrt{s}}{2}$, $|\vec{p}_1| = \sqrt{\frac{s}{4} - m_e^2}$ and $|\vec{k}_1| = \frac{\lambda^{1/2}(s, k_1^2, Q^2)}{2\sqrt{s}}$, are defined in the $p_1 + p_2$ rest frame.

The \vec{q}_1 and the $\vec{\pi}_1$ angles are generated with flat distributions

$$\phi_{q_1} = 2\pi r_{\phi_{q_1}} , \quad \cos \theta_{q_1} = -1 + 2r_{\theta_{q_1}} , \quad \phi_{\pi_1} = 2\pi r_{\phi_{\pi_1}} , \quad \cos \theta_{\pi_1} = -1 + 2r_{\theta_{\pi_1}} . \quad (53)$$

After the described changes of variables are performed, the phase space reads ($i = I$ or II)

$$d\text{Lips}_4(p_1 + p_2; q_1, q_2, \pi_1, \pi_2) = P_{s,i} dr_{k_1^2} dr_{Q^2} dr_{\theta_{k_1}} dr_{\phi_{k_1}} dr_{\theta_{q_1}} dr_{\phi_{q_1}} dr_{\theta_{\pi_1}} dr_{\phi_{\pi_1}} , \quad (54)$$

with

$$\begin{aligned}
P_{s,i} = & \frac{1}{4(4\pi)^5 s} \sqrt{1 - \frac{4m_\pi^2}{Q^2}} \sqrt{1 - \frac{4m_e^2}{k_1^2}} \lambda^{1/2}(s, Q^2, k_1^2) ((\sqrt{s} - 2m_e)^2 - 4m_\pi^2) \\
& \cdot \frac{|\vec{k}_1||\vec{p}_1|}{2k_{10}p_{10} - k_1^2} \left(\frac{-k_1^2 + 2k_{10}p_{10}}{2|\vec{k}_1||\vec{p}_1|} - \cos \theta_{k_1} \right) \left(\frac{-k_1^2 + 2k_{10}p_{10}}{2|\vec{k}_1||\vec{p}_1|} + \cos \theta_{k_1} \right) \\
& \cdot P_i \cdot \ln \left(\frac{-k_1^2 + 2k_{10}p_{10} + 2|\vec{k}_1||\vec{p}_1|}{-k_1^2 + 2k_{10}p_{10} - 2|\vec{k}_1||\vec{p}_1|} \right)^2, \tag{55}
\end{aligned}$$

where

$$P_I = \ln^3 \left(\left(1 - \sqrt{Q^2/s} \right)^2 \right) - \ln^3(4m_e^2/s), \tag{56}$$

$$P_{II} = \ln \left(\frac{(s - 4m_e^2)}{\sqrt{Q^2}(2\sqrt{s} - \sqrt{Q^2})} \right) \tag{57}$$

for the change of variables from Eq.(50) or Eq.(51) respectively. Again $0 < r_i < 1$ for $i = k_1^2, \dots, \phi_{\pi_1}$.

3.3. Merging s - and t -channel

The function, which approximates the peaking behavior of the matrix element reads

$$F = \left(\frac{1}{P(q_1, q_2)} + \frac{1}{P(q_2, q_1)} + \frac{a}{P_s} \right)^{-1}, \text{ with } P_s = \frac{P_{s,I} + bP_{s,II}}{\frac{3\ln^2(k_1^2/s)}{k_1^2} + \frac{b}{s-k_1^2}}. \tag{58}$$

Similarly to the one pion case the cross section Eq.(36) is rewritten as

$$\begin{aligned}
d\sigma_{avg}(e^+e^- \rightarrow \pi^+\pi^-e^+e^-) = & \frac{1}{4} \frac{1}{2s} \sum |\mathcal{M}_{\pi^+\pi^-}|^2 (2+a) F \times \left\{ \right. \\
& \int_0^A dr_9 dr_{Q^2} dr_{Q'^2} dr_{\theta_{q_2}} dr_{\phi_{q_2}} dr_{\theta_Q} dr_{\phi_Q} dr_{\theta_{\pi_1}} dr_{\phi_{\pi_1}} \\
& + \int_A^{2A} dr_9 dr_{Q^2} dr_{Q'^2} dr_{\theta_{q_1}} dr_{\phi_{q_1}} dr_{\theta_Q} dr_{\phi_Q} dr_{\theta_{\pi_1}} dr_{\phi_{\pi_1}} \\
& \left. + \int_{2A}^1 dr_9 \times \left[\int_0^B dr_{10} + \int_B^1 dr_{10} \right] dr_{k_1^2} dr_{Q^2} dr_{\theta_{k_1}} dr_{\phi_{k_1}} dr_{\theta_{q_1}} dr_{\phi_{q_1}} dr_{\theta_{\pi_1}} dr_{\phi_{\pi_1}} \right\}. \tag{59}
\end{aligned}$$

with $A = \frac{1}{2+a}$ and $B = \frac{P_I}{P_I+bP_{II}}$. There are three channels generated with probabilities A, A and $1 - 2A$, while the third channel branches into two channels generated with probabilities B and $1 - B$. In the first two channels the change of variables described in Section 3.1 is used, while in the third one (branched into two channels) the changes of variables are described in Section 3.2. The introduced a priori weights a and b , which give the best efficiency of the generation at the ϕ meson factory were found to be $a = 1.1$ and $b = 1000$.

The flowchart for Monte Carlo generator routine is given in Figure 4.

4. Overview of the software structure

Let us overview the directory structure of the distribution. EKHARA is distributed as a source code. The code of the Monte Carlo generator is located in the directory `ekhara-routines`. The main source file of EKHARA is `ekhara.for`. There are other source files in the directory `ekhara-routines`, which are automatically included:

- the $e^+e^- \rightarrow e^+e^-\pi^0$ mode is implemented in `routines_1pi.inc.for` and its supplementary histogramming routines are given in `routines-histograms_1pi.inc.for`;
- the $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$ mode is coded in `routines_2pi.inc.for`, its supplementary histogramming routines are in `routines-histograms_2pi.inc.for` and helicity-amplitude routines are given in `routines-helicity-aux.inc.for`;
- the routines for the matrix and vector manipulations are located in `routines-math.inc.for`;
- in `routines-user.inc.for` several routines are collected, which can be changed by a user in order to customise the operation of EKHARA; these include the data-card reading, reporting of events, form factor formulae, filling the histograms, additional phase space cuts, etc.;
- all common blocks are included from the file `common.ekhara.inc.for`. This file contains the detailed comments on the explicit purpose of the most important common variables.

The operation of the EKHARA generator requires the following steps:

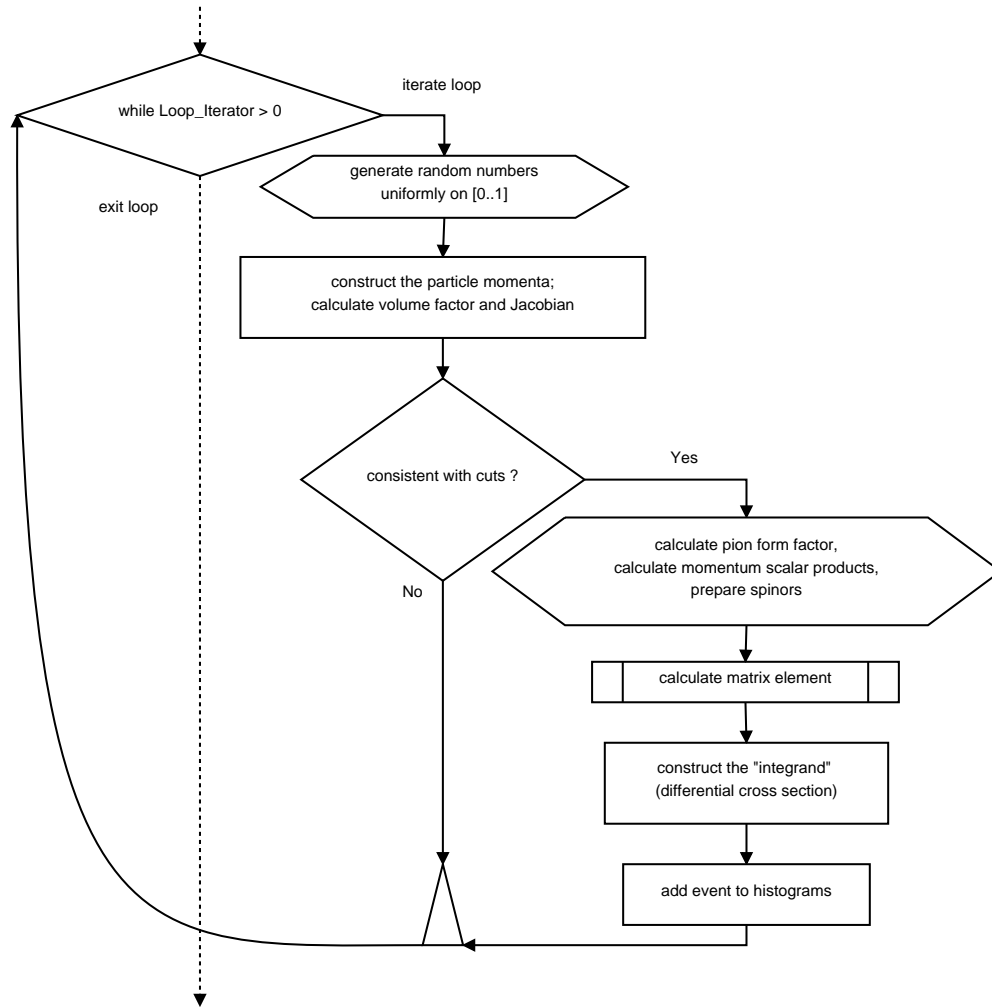


Figure 4: Flowchart for $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$ event generation.

1. initialization,
2. event generation,
3. finalization.

This structure allows us to build a stand-alone generator, as well as an interface to a separate program (e.g., a detector simulation) when EKHARA is called on an event-by-event basis. In the main directory of the distribution there are examples of both uses of EKHARA. A stand-alone example is given in `ekhara-standalone.for`. An example of an EKHARA interface on an event-by-event basis is given in `ekhara-call-example.for`. The main directory of the distributed version contains a `readme.txt` file with a short description how to compile, run and test the program in the regimes, described above. It is suggested to use the `Makefile`, which is placed in the main directory. An example of the full set of input files and the plotting environment is supplied in the `Env` subdirectory. If one uses the distributed `Makefile`, the content of the `Env` subdirectory will be put into the `EXE` subdirectory together with an executable `ekhara.exe` (for details, see Section 5 and `readme.txt`). In the following we assume that `ekhara.exe` is located for execution together with the input files in the `EXE` subdirectory.

A source code of RANLUX random number generator¹ written in C (`ranlxd.c`), together with its C-FORTRAN wrap (`ranlux_fort.c` for standard build and `ranlux_fort_vs.c` for build with `cl` in MS Windows and `xc1` in IBM AIX) are supplied in the directory `ranlux-routines`.

4.1. I/O scheme

All the input files of EKHARA are supposed to be located in the same directory as the main executable, `ekhara.exe`. There are the following types of the input files: random seeds, parameter input, data-cards and histogram settings. An example of the full set of input files can be found in the `Env` directory.

All the output files of EKHARA are written into `./output` subdirectory. There are the following types of the output files: logs of execution, histograms and events.

¹The unpublished double precision version of fast RANLUX [26, 27] written by M. Lüscher (F. James, private communication).

Input files

The main input file is called `input.dat`. It contains all global settings:

- number of generated events
- mode selection (one pion or two pions in the final state)
- histogram writing switch
- events writing switch
- particle masses and constants

Another parameter, the random seed mode switch, chooses the way the seeds for the random number generator are handled. In the constant seed mode, the file `seed.dat` is used for every execution and it is never modified. In the variable seed mode, the file `seed-v.dat` is used and on successful completion of each run, a new random seed is written into `seed-v.dat`. The latter mode is convenient for a subsequent production of statistically independent samples.

The channel-dependent parameters, which are supposed to be often changed by a user are collected in “data-cards” `card_1pi.dat` and `card_2pi.dat`. These data-cards allow to set the total energy, types of included amplitudes and kinematic cuts. A detailed description can be found in comments within these files. In `card_1pi.dat` one can also use the `piggFFsw` switch in order to select the form of the two photon pion form factor.

- 1 (WZWconst) constant form factor; this is not physical and should only be used for tests;
- 2,3,4 (rho pole, LMD, LMD+V) the form factors given in [23];
- 5 (LMD+V new) the form factor of the lowest meson dominance model with two vector resonances fitted [24] to the BABAR data [19]. This is the recommended form factor;
- 6 (quark) the form factor given in [25].

The channel-dependent histogramming settings are given in the files `histo-settings_1pi.dat` and `histo-settings_2pi.dat`.

Output: logging

The main execution log file is the `output/runflow.log`. It contains the main information about the operation mode and status of EKHARA, this information is also partly written into the standard output (i.e., the console). At the end of a successful execution, the total cross section is reported to `output/runflow.log` and also to the standard output.

A non-standard behaviour of the MC generator is reported into `output/warnings.log`, while the critical problems in the event generator operation are reported into the file `output/errors.log`.

In the case of a correct operation, the `output/errors.log` and `output/warnings.log` should remain blank. We strongly recommend to keep track on this issue and report to the authors any warnings or errors.

Output: histograms. Plotting scripts

When histogramming is allowed through settings in the `input.dat`, the plain text files with the histogram data are saved at the end of the generator execution.

- In $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$ mode the file `histograms_2pi.out` contains the data for $d\sigma/dQ^2$ histogram. One may use the plotting script `doplots.sh` from directory `histo-plotting_2pi` in order to plot this histogram (an installed Gnuplot is required).
- In $e^+e^- \rightarrow e^+e^-\pi^0$ mode there is a wide set of histograms stored in the files `histo<Number>.<variable>.dat`, where `<Number>` stands for the histogram number and `<variable>` is the histogramming variable acronym.

One may use the plotting script `do-everything.sh` in the directory `histo-plotting_1pi` in order to plot all the histograms and collect them into a single postscript file. An installed L^AT_EX system is required for the latter.

One may use the plotting script `doplots.sh` in the directory `t1-t2-bars_1pi` in order to plot the 3D-bar graph, which shows the distribution in two variables: t_1 and t_2 .

As the histograms are stored as plain text files the user can use also her/his favourite plotting programs to visualize the histograms.

Output: events

The generated four-momenta of the particles are stored in the following variables, which can be accessed through common blocks:

p1	initial positron,
p2	initial electron,
q1	final positron,
q2	final electron,
qpion	final pseudoscalar ($e^+e^- \rightarrow e^+e^-\pi^0$ mode),
pi1, pi2	final pseudoscalars ($e^+e^- \rightarrow e^+e^-\pi^+\pi^-$ mode).

In the file `ekhara-call-example.for` we give an example how the generated momenta can be used, when EKHARA works in the event-by-event regime. In the standalone regime we suggest to use the routine `reportevent_1pi` defined in the file `routines-user.inc.for`, which is called automatically for every accepted unweighted event ($e^+e^- \rightarrow e^+e^-\pi^0$ mode only). In the distributed version this routine writes the events to the file `output/events.out` when `WriteEvents` flag is on. One can modify this routine in order to accommodate the way how the generated events are collected.

4.2. Selected procedures

The top-level interface to the Monte Carlo generator is provided by the routine

EKHARA(i)	i = -1: initialize,
	i = 0: generate event(s),
	i = 1: finalize.

Only this routine should be called from an external program, when you use EKHARA in the event-by-event regime, see example `ekhara-call-example.for`.

In order to describe briefly the “internal” structure of EKHARA, we list several important routines.

EKHARA_INIT_read	reading the input files and datacards,
EKHARA_INIT_set	initialization of the MC loop and mappings,
EKHARA_RUN	MC loop execution,
EKHARA_FIN	MC finalization, saving the results.

$e^+e^- \rightarrow e^+e^-\pi^0$ procedures

<code>mc_loop_1pi</code>	Monte Carlo loop (see the flowchart in Figure 3),
<code>EvalUpperBound</code>	evaluation of the upper bound for the Monte Carlo integrand,
<code>phasespace_1pi</code>	a wrap for the phase space generation routines,
<code>eventselection_1pi</code>	kinematic cuts,
<code>m_el_1pi</code>	calculation of the matrix element for $e^+e^- \rightarrow e^+e^-\pi^0$,
<code>tellSIGMA_1pi</code>	reports the total cross section.

$e^+e^- \rightarrow e^+e^-\pi^+\pi^-$ procedures

<code>mc_loop_2pi</code>	Monte Carlo loop (see the flowchart in Figure 4),
<code>eventselection_2pi</code>	kinematic cuts,
<code>phsp1</code>	phase space generation routine (branch 1),
<code>phsp2</code>	phase space generation routine (branch 2),
<code>phsp3</code>	phase space generation routine (branch 3),
<code>matrixelmt</code>	calculation of the matrix element for $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$,
<code>tellSIGMA_2pi</code>	reports the total cross section.

For details see the comments in the source code.

5. Compilation instructions

Being distributed as a source code the program does not require installation, but a compilation and linking is needed. EKHARA does not need any specific external libraries. In order to compile the program, a user may run any OS (UNIX, Linux, MS Windows, etc) with correctly installed

- FORTRAN 77 compiler with support of quadruple precision,
- C compiler.

The program was tested on the following platforms:

- Linux (Ubuntu 8.04.3)
GNU C Compiler (gcc) 4.2.4
Intel(R) FORTRAN Compiler (ifort) 11.0.20080930

Description	Linux	Windows	IBM AIX
Default: Standalone MC generator, sample input files and histogramming routines. Everything put into EXE directory	default	default-vs	default-ibm
Compile everything: default, ranlux-testing program and seed-production programs	all	all-vs	all-ibm
Testrun: Compile everything and execute the testrun scripts	test	test-vs	test-ibm
Remove the redundant and temporary files	clean	clean-vs	clean-ibm

Table 7: The main **Makefile** targets

- MS Windows (XP SP3)
MS Visual Studio 2008 (nmake, cl)
Intel(R) FORTRAN Compiler (ifort) 11.

The program distribution contains the **Makefile**, with targets for Linux, IBM AIX and Windows environments. The **Makefile** is annotated, in order to help a user to tune it up for the own requirements. The main **Makefile** targets are listed in Table 7.

For example, in Linux, a simple way to compile a program is to issue **make default**, being in the directory where the **Makefile** is located. This will produce **ekhara.exe** (main program executable) and copy it into the sub-directory **EXE**, together with the contents of **Env** sub-directory. The latter contains the set of sample input files and histogram plotting scripts. We provide a full set of necessary input files in the distribution package. It is advised to execute **ekhara.exe** in the directory **EXE**, where it is placed by default. Every time one does **make default**, the input files in **EXE** are replaced with the sample ones from **Env**.

In order to produce only an object file with the EKHARA MC generator, one can use, for example

```
ifort -c ekhara-routines/ekhara.for -o ./ekhara.o
```

EKHARA needs a random seed for operation. Different random seeds can be obtained by using a Makefile target `seed_prod-ifort`. It produces an executable program `seed_prod.exe`, which generates a set of random seeds.

6. Test run description

It is recommended to test the random number generator on a given machine, before using EKHARA. It is also important to check whether EKHARA can function properly on a given operational system and that there are no critical bugs due to the compiler. We provide a testrun package for these purposes.

It is suggested to use the Makefile targets `test`, `test-vs` or `test-ibm` depending on your environment (Linux, Windows and IBM AIX, correspondingly). This will automatically prepare and execute the following two test steps.

The first step of the testrun is the random number generator control. The source file `testlxf.for` contains the `ranlux` test routines.

The second step is the verification if the user-compiled EKHARA can reproduce the set of results, created by a well-tested copy of EKHARA in various modes. The testrun environment contains directories `test` and `test-vs` with precalculated data for a comparison, along with the random seed and input files for each mode. The script `test.sh` is responsible for the execution of a user-compiled `ekhara.exe` in all the control modes and for the comparison of the output.

Please read carefully the output of the testrun execution in your console and be sure there are no warnings and/or error messages.

7. Customization of the source code by a user

We leave for a user an option to customise the generator to her/his needs by editing the source code file `ekhara-routines/routines-user.inc.for`. Notice, we always use explicit declaration of identifiers and the `implicit none` statement is written down in each routine.

In the file `ekhara-routines/routines-user.inc.for` one can change

- the data-card reading (routines `read_card_1pi` and `read_card_2pi`),
- the form-factor formula (routine `piggFF`),
- the events reporting (routine `reportevent_1pi`),
- histogramming (routines `histo_event_1pi` and `histo_event_2pi`),
- additional kinematic cuts (routines `ExtraCuts_1pi` and `ExtraCuts_2pi`).

8. Validation of the generator

In Fig. 2 we demonstrated an agreement of the Monte Carlo simulation of $e^+e^- \rightarrow e^+e^-\pi^0$ in the “single-tag” mode with the experimental data from CLEO [18] and BaBar [19]. We conclude that the matrix element is well under control and the applied pion transition form factor (LMD+V) [24] is in agreement with data. The procedure of the phase space generation was validated by means of the high statistics phase space volume calculation in EKHARA and comparison of the result with that from the independent dedicated numerical calculation. The volume was also compared to that obtained by GALUGA generator [20]. We have also verified that our numerical results for the three-body phase space volume in the limit of massless π^0 reproduce well those of the analytic expression.

The $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$ mode is validated by means of the reproduction of the results from the previous version of EKHARA. In [1, 2, 3] the tests of this part are presented in detail.

9. Summary

An update (version 2.0) of the Monte Carlo event generator EKHARA is presented. It generates processes $e^+e^- \rightarrow e^+e^-\pi^0$ and $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$. The newly added channel ($e^+e^- \rightarrow e^+e^-\pi^0$) is important for $\gamma^*\gamma^*$ physics and can be used for the pion transition form factor studies at meson factories.

Acknowledgements

We would like to thank Fred Jegerlehner and Andreas Nyffeler for discussion of the physics case of $\gamma^*\gamma^* \rightarrow \pi^0$, Vladimir Druzhinin for drawing our attention to Ref. [20], Danilo Babusci, Dario Moricciani and Graziano

Venanzoni for discussion of the experimental project KLOE-2 and simulation issues. We are grateful to Germán Rodrigo for his kind hospitality at Instituto de Física Corpuscular CSIC (Valencia) and to Achim Denig at Institut für Kernphysik J.Gutenberg-Universität (Mainz), where a part of this work has been done. This work was partially supported by MRTN-CT-2006-035482 “FLAVIANet” under the Sixth Framework Program of EU, Polish Ministry of Science and High Education from budget for science for years 2010-2013: grant number N N202 102638 and the European Community-Research Infrastructure Integrating Activity “Study of Strongly Interacting Matter” (acronym HadronPhysics2, Grant Agreement n. 227431) under the Seventh Framework Program of EU.

References

- [1] H. Czyz, E. Nowak-Kubat, Radiative return via electron pair production: Monte Carlo simulation of the process $e^+e^- \rightarrow \pi^+\pi^-e^+e^-$, *Acta Phys. Polon. B36* (2005) 3425–3434.
- [2] H. Czyz, E. Nowak-Kubat, The reaction $e^+e^- \rightarrow e^+e^-\pi^+\pi^-$ and the pion form factor measurements via the radiative return method, *Phys. Lett. B634* (2006) 493–497.
- [3] H. Czyz, E. Nowak, $e^+e^- \rightarrow \pi^+\pi^-e^+e^-$: A potential background for $\sigma(e^+e^- \rightarrow \pi^+\pi^-)$ measurement via radiative return method, *Acta Phys. Polon. B34* (2003) 5231–5238.
- [4] A. Aloisio, et al., Measurement of $\sigma(e^+e^- \rightarrow \pi^+\pi^-\gamma)$ and extraction of $\sigma(e^+e^- \rightarrow \pi^+\pi^-)$ below 1-GeV with the KLOE detector, *Phys. Lett. B606* (2005) 12–24.
- [5] F. Ambrosino, et al., Measurement of $\sigma(e^+e^- \rightarrow \pi^+\pi^-\gamma(\gamma))$ and the pion contribution to the muon anomaly with the KLOE detector, *Phys. Lett. B670* (2009) 285–291.
- [6] G. Amelino-Camelia, et al., Physics with the KLOE-2 experiment at the upgraded DAΦNE, *Eur. Phys. J. C68* (2010) 619–681.
- [7] S. Actis, et al., Quest for precision in hadronic cross sections at low energy: Monte Carlo tools vs. experimental data, *Eur. Phys. J. C66* (2010) 585–686.

- [8] D. M. Asner, et al., Physics at BES-III (2008).
- [9] D. Babusci, et al., The Low Energy Tagger for the KLOE-2 experiment (2009).
- [10] A. Courau, A fast Monte Carlo generator for $e^+e^- \rightarrow e^+e^-X$ untagged experiments (1984). SLAC-PUB-3363.
- [11] G. Alexander, et al., Two photon physics capabilities of KLOE at DAPHNE, *Nuovo Cim.* A107 (1994) 837–862.
- [12] S. Bellucci, A. Courau, S. Ong, Azimuthal correlations in $\gamma\gamma \rightarrow \pi^0\pi^0$ at DAPHNE (1994). LNF-94-078-P.
- [13] S. Ong, Two-photon reactions with KLOE detector at DAPHNE (1999).
- [14] F. Nguyen, F. Piccinini, A. D. Polosa, $e^+e^- \rightarrow e^+e^-\pi^0\pi^0$ at DAFNE, *Eur. Phys. J.* C47 (2006) 65–70.
- [15] S. Uehara, TREPS: A Monte Carlo event generator for two-photon processes at e^+e^- colliders using an equivalent photon approximation (1996). KEK-REPORT-96-11.
- [16] T. Mori, et al., High statistics measurement of the cross sections of $\gamma\gamma \rightarrow \pi^+\pi^-$ production, *J. Phys. Soc. Jap.* 76 (2007) 074102.
- [17] T. Mori, et al., High statistics study of $f_0(980)$ resonance in $\gamma\gamma \rightarrow \pi^+\pi^-$ production, *Phys. Rev.* D75 (2007) 051101.
- [18] J. Gronberg, et al., Measurements of the meson photon transition form factors of light pseudoscalar mesons at large momentum transfer, *Phys. Rev.* D57 (1998) 33–54.
- [19] B. Aubert, et al., Measurement of the $\gamma\gamma^* \rightarrow \pi^0$ transition form factor, *Phys. Rev.* D80 (2009) 052002.
- [20] G. A. Schuler, Two-photon physics with GALUGA 2.0, *Comput. Phys. Commun.* 108 (1998) 279–303.
- [21] F. A. Berends, R. van Gulik, GaGaRes: A Monte Carlo generator for resonance production in two-photon physics, *Comput. Phys. Commun.* 144 (2002) 82–103.

- [22] S. J. Brodsky, T. Kinoshita, H. Terazawa, Two Photon Mechanism of Particle Production by High-Energy Colliding Beams, Phys. Rev. D4 (1971) 1532–1557.
- [23] M. Knecht, A. Nyffeler, Hadronic light-by-light corrections to the muon g-2: The pion-pole contribution, Phys. Rev. D65 (2002) 073034.
- [24] A. Nyffeler, Hadronic light-by-light scattering in the muon g-2: a new short-distance constraint on pion exchange (2009).
- [25] A. E. Dorokhov, How the recent BABAR data for $P \rightarrow \gamma\gamma^*$ affect the Standard Model predictions for the rare decays $P \rightarrow l^+l^-$ (2009).
- [26] M. Luscher, A Portable high quality random number generator for lattice field theory simulations, Comput. Phys. Commun. 79 (1994) 100–110.
- [27] F. James, RANLUX: A FORTRAN implementation of the high quality pseudorandom number generator of Luscher, Comp. Phys. Commun. 79 (1994) 111–114.